

REMARKS

This Amendment is filed in response to the FINAL Office Action mailed on July 8, 2009. A Request for Continued Examination and the associated fee is also filed herewith. All objections and rejections are respectfully traversed.

Claims 1-8, 12, 14-25, 29, 31-35, 37-39, 41-42, 46-50 are in the case.

No new claims have been added.

Claims 1, 3, 6-8, 12, 14-18, 20, 22-23, 25, 29, 31-35, 37-39, 41-42, and 47-50 have been amended to better claim the invention.

Interview Summary

Applicant would like to thank Examiner Lovel for conducting the Applicant Initiated Interview on October 21, 2009 and for helping to advance this Application closer to allowance. Generally, as will be elaborated upon in greater detail below, the issue discussed involved Applicant's use of a **management application configured to scan the root of each PCPI comprising the directory tree to generate the index associated with the directory tree**. Specifically, Applicant discussed how the cited prior art references (i.e., Prahlad, Shi, and Arakawa) broadly teach indexing, e.g., a full snapshot of all files (e.g., of a file system) at a single point-in-time, whereas Applicant claims, broadly speaking, generating an index of different versions of a particular file (e.g., *associated with a particular directory tree*) created at different points-in-time. Examiner stated that if Applicant's contentions were accurate, the claims would overcome the cited prior art references. Examiner noted that a closer look at the references would be required to verify Applicant's contentions and that another search would be required. Examiner is encouraged to contact the undersigned attorney with any questions.

Rejections Under 35 U.S.C. §103

At paragraph 6 of the Office Action, claims 1, 2, 17-19, 34-35, 38-39, and 46-50 were rejected under 35 U.S.C. §103(a) as being unpatentable over Prahlad et al., U.S.

Patent Application Publication No. 2004/00100487 (hereinafter “Prahlad”), in view of Shi et al., U.S. Patent Application Publication No. 2003/0105889 (hereinafter “Shi”).

The present invention, as set forth in representative claim 1, comprises in part:

1. A system for indexing and manipulating backup data stored on a destination storage system, comprising:
 - one or more source servers configured to transmit the backup data to the destination storage system;
 - a management application executed by a processor, wherein the **management application is configured to** (a) communicate with the destination storage system and further configured to access data identifiers related to the backup data organized in a directory tree structure representing a plurality of persistent consistency point images (PCPIs) of the backup data, wherein each PCPI is associated with a creation time, (b) **scan a root of each PCPI comprising the directory tree to generate an index** of directories, files, or qtrees *associated with the directory tree*, and (c) organize the data identifiers into a structure that enables the backup data to be displayed; and
 - a user interface to select a directory, file, or qtree to view, wherein the management application is further configured to return a list of the selected directory, file, or qtree and one or more versions of the selected directory, file, or qtree.

Prahlad teaches managing snapshots [0055]. These snapshots may be indexed by providing point-in-time browsing and point-in-time managing of snapshot images [0055]. Specifically, a user enters a point-in-time of which to browse the snapshots [0067]. In other words, Prahlad teaches indexing and browsing snapshots related to a specific point-in-time.

Similarly, Shi teaches restoring a selected application to a prior configuration identified by a particular date [0056]. Specifically, if an application is causing problems for a user, a window may be opened with a list of applications. The user selects the application causing the problems, and selects a prior point-in-time configuration of the application.

Applicant respectfully urges that Prahlad, taken singly or in any combination with Shi, does not disclose Applicant's claimed novel and non-obvious use of a **management application configured to scan the root of each PCPI comprising the directory tree to generate the index associated with the directory tree.**

Applicant claims, in part, a **management application configured to** communicate with a destination storage system and further configured to access data identifiers related to the backup data organized in a directory tree structure representing a plurality of persistent consistency point images (PCPIs) of the backup data. Each PCPI may pertain to a particular directory tree or, e.g., file. In other words, each PCPI of that particular directory tree may also comprise that directory tree. Notably, as more PCPIs of the particular directory tree are created at different points-in-time, each of those multiple PCPIs also comprise that directory tree. With that being said, there may be multiple PCPIs required to show a full image of, for example, an active file system at a previous point in time. However, not every PCPI necessarily comprises the same directory tree. It may be desirable to generate an index to examine all PCPIs associated with a specific directory tree spanning multiple points-in-time (i.e., **each PCPI comprising the directory tree**), rather than generating an index to examine every PCPI making up a single point-in-time image of the entire active file system. In other words, where information on a given directory is requested, the full range of PCPIs containing that directory (e.g. the root of each relevant PCPI comprising that directory tree) may be located and read to generate an index associated with the specific directory tree. Again, this is in contrast to requesting information on all directories at a single point-in-time. Applicant accomplishes this, in part, by using a **management application configured to scan the root of each PCPI comprising the (requested) directory tree to generate the index associated with the directory tree.**

Applicant respectfully argues that Prahlad does not teach Applicant's claimed novel **management application configured to scan the root of each PCPI comprising**

the directory tree to generate the index associated with the directory tree. Specifically, Prahlad teaches indexing all PCPIs at a single point-in-time. However, any PCPIs of, e.g., the file system, generated at different points-in-time are not included in Prahlad's index. This is because Prahlad teaches indexing all snapshots associated with a particular point-in-time, so any snapshots taken at a different point-in-time is not indexed. In contrast, Applicant claims a management application configured to scan the root of *each PCPI comprising the directory tree to generate the index associated with the directory tree*. Put another way, broadly stated, Applicant claims indexing PCPIs taken at different times, wherein those PCPIs all pertain to the same directory tree (e.g., file). In other words, Prahlad teaches indexing all snapshots of all directories that were generated at a specific point-in-time, whereas Applicant claims indexing multiple PCPIs of a specific directory tree that were generated at different times. As such, because Prahlad does not teach indexing snapshots associated with a particular directory tree at different points-in-time, Prahlad is silent to Applicant's claimed novel **management application configured to scan the root of each PCPI comprising the directory tree to generate the index associated with the directory tree**.

Similarly, Applicant respectfully argues that Shi is silent to Applicant's claimed novel **management application configured to scan the root of each PCPI comprising the directory tree to generate the index associated with the directory tree**. Specifically, similarly to Prahlad, Shi also teaches displaying prior configurations, albeit of an application, based on a specific point-in-time (i.e., date). In contrast, Applicant claims a management application configured to scan the root of *each PCPI comprising the directory tree* to generate the index associated with the directory tree. Thus, because Shi does not teach indexing snapshots associated with a particular directory tree at different points-in-time, Shi is silent to Applicant's claimed novel **management application configured to scan the root of each PCPI comprising the directory tree to generate the index associated with the directory tree**.

Finally, it should be noted that neither Prahlad nor Shi disclose generating an index associated with a particular directory tree by a management application configured to **scan the root of each PCPI comprising the directory tree**. In contrast, Applicant claims a management application configured to **scan the root of each PCPI comprising the directory tree** to generate the index associated with the directory tree. As such, because neither Prahlad nor Shi disclose scanning the roots of each PCPI associated with a particular directory tree to generate the index, both Prahlad and Shi must be silent to Applicant's claimed novel **management application configured to scan the root of each PCPI comprising the directory tree to generate the index associated with the directory tree**.

Accordingly, Applicant respectfully urges that Prahlad, taken singly or in any combination with Shi, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. §103(a). Prahlad and Shi, taken singly or in any combination, do not disclose Applicant's claimed novel and non-obvious use of **management application configured to scan the root of each PCPI comprising the directory tree to generate the index associated with the directory tree**.

One or more claims rejected by Examiner at paragraph 6 are dependent claims that are dependent from independent claims which are believed to be allowable for the reasons described above. Accordingly, those dependent claim(s) rejected by Examiner at paragraph 6 are believed to be in condition for allowance.

At paragraph 7 of the Office Action, claims 3-6, 20-23, 37, and 41 were rejected under 35 U.S.C. §103(a) as being unpatentable over Prahlad, and in view of Shi, and in further view of Armangau, U.S. Patent No. 6,434,681 (hereinafter "Armangau").

Applicant respectfully notes that claims 3-6, 20-23, 37, and 41 are dependent claims that depend from independent claims believed to be in condition for allowance. Accordingly, claims 3-6, 20-23, 37, and 41 are believed to be in condition for allowance.

At paragraph 8 of the Office Action, claims 7, 8, 12-16, 24-33, and 42 were rejected under 35 U.S.C. §103(a) as being unpatentable over Prahlad, and in view of Shi, and in further view of Arakawa et al., U.S. Patent Application Publication No. 2003/0131207 (hereinafter “Arakawa”).

The present invention, as set forth in representative claim 42, comprises in part:

42. A system, comprising:

a source storage system configured to generate a plurality of persistent consistency point images (PCPIs) associated with a particular directory tree, and further configured to transfer the plurality of PCPIs to a destination storage system;

the destination storage system configured to execute a management client, wherein the management client is configured to organize the plurality of PCPIs into an index using a database to allow the plurality of PCPIs to be displayed in (a) a **listing of source data entries indexed by the particular directory tree, wherein each PCPI of the particular directory tree is created at one or more different times** (b) a listing of source data entries indexed by names of the source storage system, and (c) a listing of source data entries indexed by names of volumes of the destination storage system in which backup data from the source storage system resides; and

an interface configured to select a data entry for the particular directory tree, and **the management client further configured to return a list of the plurality of PCPIs associated with the particular directory tree.**

Arakawa teaches a virtualized volume snapshot formation method. However, Arakawa is silent to Applicant's claimed novel **listing of source data entries indexed by the particular directory tree, wherein each PCPI of the particular directory tree is created at one or more different times and listing of source data entries indexed by the particular directory tree, wherein each PCPI of the particular directory tree is created at one or more different times.**

Additionally, as noted above, Prahlad and Shi also do not disclose or teach Applicant's claimed novel **management application configured to scan the root of each PCPI comprising the directory tree to generate the index associated with the directory tree**. As such, because claim 42 comprises similar limitations not shown by either prior art reference (i.e., indexing PCPIs of a particular directory tree generated at different points-in-time), Applicant respectfully urges that Prahlad, taken singly or in any combination with Shi and/or Arakawa, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. §103(a). Specifically, Prahlad and/or Shi and/or Arakawa, taken singly or in any combination, do not disclose Applicant's claimed novel and non-obvious use of **listing of source data entries indexed by the particular directory tree, wherein each PCPI of the particular directory tree is created at one or more different times and listing of source data entries indexed by the particular directory tree, wherein each PCPI of the particular directory tree is created at one or more different times**.

One or more claims rejected by Examiner at paragraph 8 are dependent claims that are dependent from independent claims which are believed to be allowable for the reasons described above. Accordingly, those dependent claim(s) rejected by Examiner at paragraph 8 are believed to be in condition for allowance.

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art was derived, in part, from the following excerpts:

Prahlad

[0055] The volume snapshot service 505 is either a software snapshot application from a software snapshot provider, such as Microsoft NET Server, or a hardware snapshot application from a hardware snapshot provider, such as EMC or Compaq. The software snapshot image applications will, in one embodiment, create a space efficient copy that is exposed as a separate logical volume using a copy-on-write technique.

Hardware snapshot image applications, typically accompanied with a RAID cabinet, create a mirror or clone copy of application data or primary volumes. Once the volume snapshot service 505 has taken the snapshot image, the snapshot data is passed to the snapshot manager 503, which indexes the snapshot image enabling snapshot management. Indexing generally denotes associating snapshots with information that may be useful in managing snapshots, such as the date the snapshot was created, the lifespan of the snapshot, etc. Managing generally includes, but is not limited to, copying, deleting, displaying, browsing, changing properties, or restoring the snapshots or data therein. **Indexing generally provides point-in-time browse and management**, such as recovery, capability of the snapshot images and of the quick recovery volumes. Users can choose to persist or retain snapshot images well beyond the lifetime of the requesting application or module. The snapshot manager 503 may then communicate the snapshot data to the quick recovery agent 108 for copying to the quick recovery volume 118, or to the media agent 106 for copying to the archival storage unit 112.

[0067] In one embodiment, users may specify a **point-in-time** for which browsing and restoration may occur. In that instance, **the browser application determines if there are any existing snapshot images present as of the point-in-time specified**. Snapshot data found to be available as of the point-in-time the user specified is displayed to the user. Snapshot data is displayed if it exists and qualifies as valid data as of the point-in-time. If the browser application does not find a snapshot, backup copies, such as primary copies, and secondary copies, and quick recovery volumes are presented or accessed for data recovery or restoration. If the user chooses to drill down a given snapshot image, quick recovery volume, or backup copy to see the contents therein, such as by selecting or double-clicking an item, the item is displayed at the requesting client computer in an appropriate user interface screen, such as in an interface screen provided by the application associated with the item. Association generally refers to the relationship between a file and the application that created it. (emphasis added)

Shi

[0055] Once an application has been selected by the user, the application restore tool displays a user interface window 276 that informs the user that the selected application will be restored to the prior configuration **identified by date**. If this meets the user needs, the user need only select the ok button 278 to restore the application to that prior configuration. (emphasis added)

[0056] If, however, this prior configuration does not meet the needs of the user, the user may select the advanced options button 280 for more options. When this button is selected, a user interface 282 such as that illustrated in FIG. 8 is displayed. In this user interface 282 all of the available configurations for the selected application are displayed for the user selection in a window 284. Each of these configurations relate to a prior bind history "snapshot" that was assembled during a prior configuration of the application. Additionally, an option is provided so that the user may restore the application to the original configuration, or to an application safe mode that allows individual binds to particular versions of the assemblies required for operation of the application. This takes the user back to the state where the assemblies used by the application are the versions that were originally intended by the application developer (ie. avoids applying publisher policy). Once one of the application configurations is selected, the user simply selects the apply button 286 to restore the application to a previous configuration. A final user interface 288 illustrated in FIG. 9 may be displayed to the user to provide additional instructions regarding the restore process. This tool will then ensure that proper programmatic steps can be executed to create configuration files that describe the new binding policy redirect information to force the application to use the intended version of the assemblies required by that application.

Conclusion

All new claims and/or claim amendments are believed to be fully supported by Applicant's specification.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account
No. 03-1237.

Respectfully submitted,

/Michael T. Abramson/
Michael T. Abramson
Reg. No. 60,320
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927